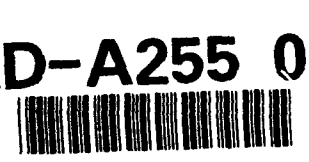


(2)

marble

Building Mission Critical Decision Support Systems

The *Decision Builder Tool Kit*™ and the Marble Workgroup Computing Paradigm

a Final Report submitted to the Office of Naval Research under Contract N00014-90-C-0102

by
Marble Associates, Inc.
38 Edge Hill Road
Waltham, Massachusetts U.S.A. 02154
telephone: (617) 891-5555

April 1, 1992

10 123

Building Mission Critical Decision Support Systems

The *Decision Builder Tool Kit*™ and the Marble Workgroup Computing Paradigm

Marble Associates, Inc.
April 1, 1992

A custom decision support system is the preferred solution for most mission-critical, information-intensive tasks throughout industry. Until recently, constructing such custom systems was prohibitively time consuming and error-prone. Tools such as Marble's ***Decision Builder***™ promise to reduce the development time and improve the effectiveness of custom decision aids. This White Paper highlights ***Decision Builder***'s inferencing technology, open architecture, graphic tools, and network orientation. It also suggests a new vision for computer-based decision aids across industries in light of ***Decision Builder*** and Marble's ***Workgroup Computing Paradigm***.

1.0 THE NEED FOR CUSTOM DECISION SUPPORT SYSTEMS

Over the past ten years computers have permeated the workplace. Mass produced decision aids such as spreadsheets and constraint languages have been embraced by users throughout industry. These tools have increased productivity and enabled complex undertakings. Although spreadsheets and other software products are good at automating the processes that are common across all industries, those functions that are *particular* to a given industry – especially those that create competitive advantage – seldom benefit. In the age of desktop computers, limits on executive productivity are often defined by the failure of mass produced applications to support the decision making needs of a particular work place.

Custom applications have always been the choice for addressing the limitations of generic software. However, the traditionally large investment of time and other resources needed to create a mission critical decision support system (*i.e.*, one that creates competitive advantage) has made this solution impractical for most organizations.

2.0 THE NEED FOR NEW TECHNOLOGY IN AN OLD FIELD

Advances in hardware technology over the last decade have far outpaced advances in mass produced software tools. A typical example of this is that the popular constraint language TK! Solver™ running on 64Mb, 40 MIPS Sparcstation today is essentially the same program as TK! Solver was eight years ago when it ran on a 64Kb, 0.01 MIPS Apple II. The first spreadsheet appeared in the late 1970s, but until this year, no VisiCalc™ user would have found any major surprises in cell naming conventions in any spreadsheet on the market.

There are many reasons for this. Pressures on a software vendor to retain old, tested code, to support the lowest common denominator in a hardware platform, and to avoid confusing old customers lead to applications that fail to take advantage of the exceptional features of a new machine. The rare exceptions to this trend are found in applications such as Lotus Improv™ and ***Decision Builder***: these products offer the important benefits of emerging technology while integrating with the existing computing environments of most organizations.

3.0 BUILDING A CUSTOM DECISION SUPPORT SYSTEM (DSS)

A custom DSS fills a mission-critical need by fostering an organization's competitive position and/or operational efficiency.

All too often, however, these systems fail. They do not earn or sustain an acceptable return on the substantial investments required to build them. In many cases the problem is ill-defined, or difficult to bound, or mutable; consequently, the decision support system produces irrelevant or inaccurate results. In others, the computer-based decision aids fail to account for changing computing environments. Instead of integrating with emerging organizational internetworks,[1] they are isolated from the main "computing concourse." Thus isolated, they cannot tap networked resources – databases of trend data, real-time data feeds and sensors; *etc.* – to provide accurate and time-sensitive decision support.

The failures of past DSS initiatives suggest some new guidelines for organizations that undertake DSS initiatives:

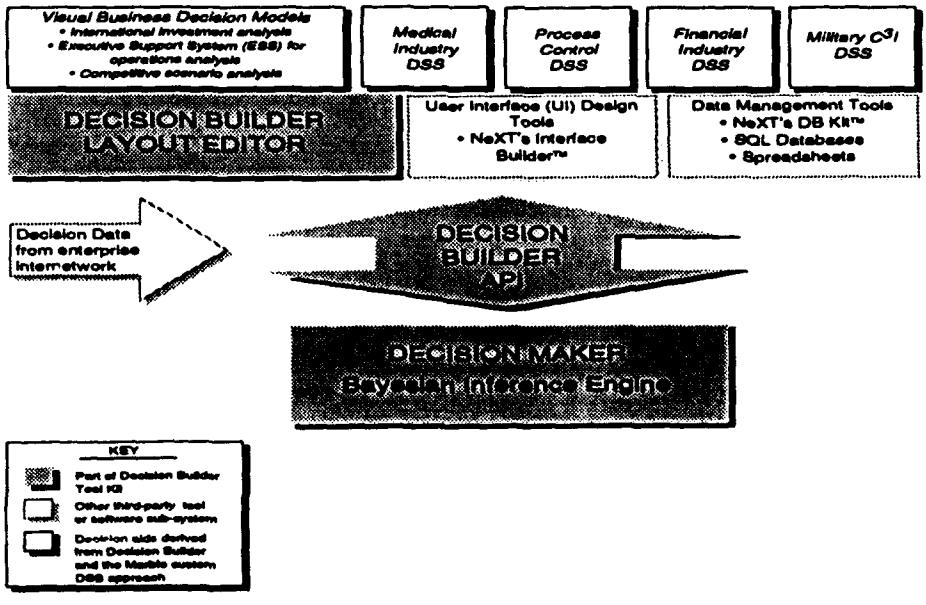
1. *identify needs and required benefits* – A custom DSS should focus on *real work* and *real issues*, and it should provide tangible benefits.
2. *select appropriate tools* – Custom DSS developers should leverage mature, well-tested algorithms (*e.g.*, those embedded in a spreadsheet or inference engine), software sub-systems (*e.g.*, distributed databases), and hardware and networks (*e.g.*, multimedia workstations, compute servers, TCP/IP-based networks and network devices) without having to write, test, debug, and optimize all elements of the DSS from scratch.
3. *deliver prototype systems early* – Through prototyping, a DSS developer can clarify the decision-making process and continually refine the system to focus on key issues. The prototyping process should lead to *working systems*, so that investments in DSS technology can be evaluated early.

4.0 DECISION BUILDER AND THE MARBLE APPROACH

Marble orients its Workgroup Computing Paradigm, rapid prototyping capability, and the Decision Builder Tool Kit to focus on the three-phase process outlined above.

1. The term "internetwork" refers to the distributed computing model that has been adopted by many organizations. This distributed model is characterized by many workgroups and their Local Area Networks, inter-connected to other workgroups and enterprise data via a high-speed network backbone. The Marble Workgroup Computing Paradigm is a useful guide in designing and building distributed solutions.

FIGURE 1. Decision Builder and the "tool integration" approach to delivering decision support systems



4.1 IDENTIFYING NEEDS

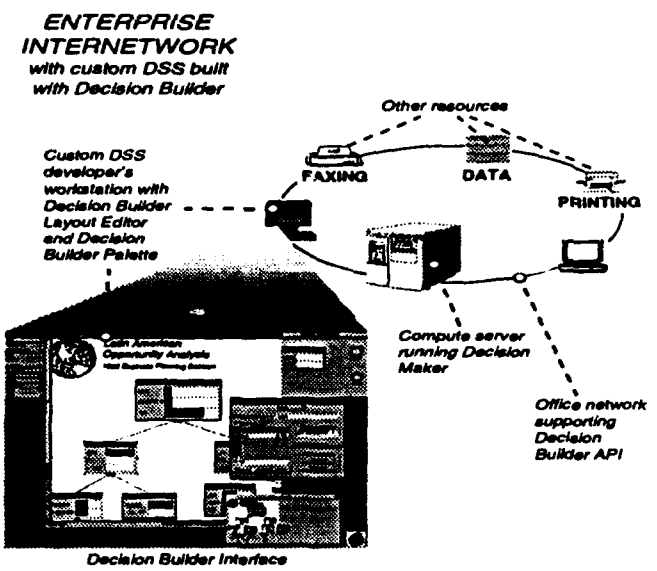
The Marble Workgroup Computing Paradigm and strategic information technology (IT) consulting process has been applied in various contexts and industries to develop business-driven IT solutions. The Workgroup Computing Paradigm addresses the dissatisfaction many executives experience due to insufficient productivity gains from technology investments. Fundamentally, Workgroup Computing recognizes that work is accomplished in *groups*, not by individuals alone; therefore, computing investments should transcend individual productivity benchmarks. The Workgroup Computing Paradigm asserts that:

- technology should focus on real work, and therefore, the *workgroup*;
- in any enterprise, relevant information and knowledge should be both *accessible* (to any workgroup) and *secure*; and
- the granularity of the workgroup computing model mirrors the business structure of an enterprise.

The deliverable under the Workgroup Computing Paradigm is a practical, enterprise-specific strategy for building distributed computing solutions. These solutions redeploy computing assets according to fluid organizational structures, and they usually tie local workgroup networks (composed of heterogeneous desktop computers, various peripherals, data/file/compute servers) to other workgroups via an enterprise backbone.

In crafting a custom decision support system, the Workgroup Computing Paradigm serves as a methodological anchor for hashing through all important business and technical issues. It recognizes the importance of distributed data and applications in the 1990s, as organizations shift to meet competitive pressures and take advantage of global opportunities. The resulting production DSS will thrive in an enterprise internetwork, as suggested in Figure 2.

FIGURE 2. A custom DSS built with Decision Builder for a distributed heterogeneous corporate environment.

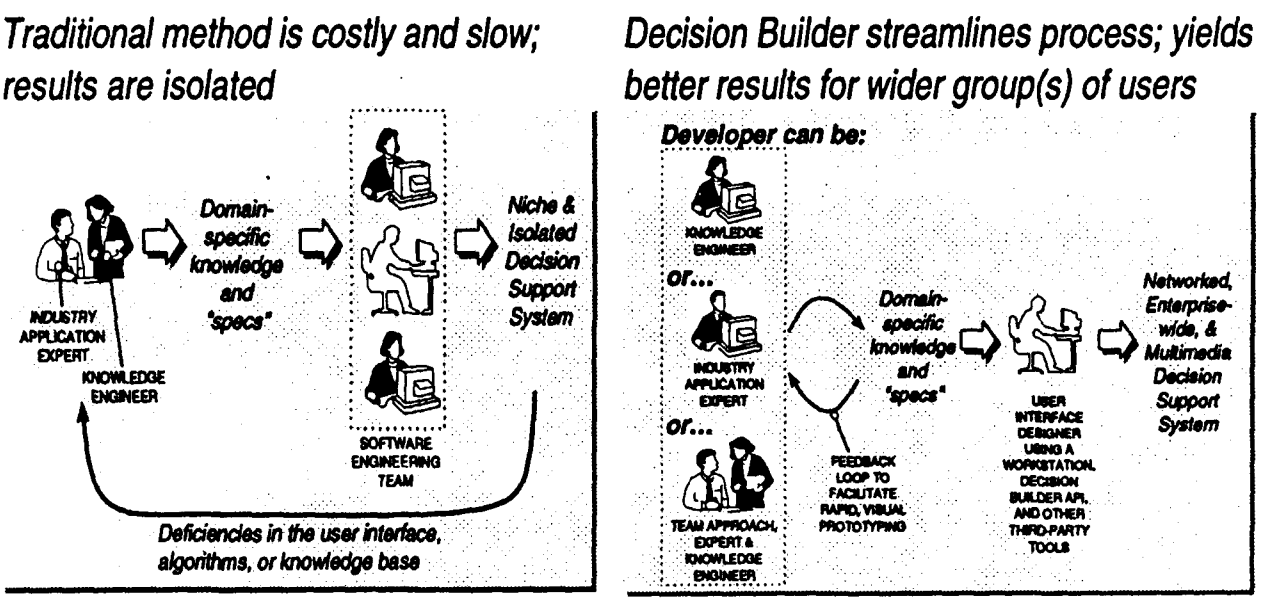


Marble leverages the *tools it has developed over the past* ten years – including the Decision Builder layout editor – to clarify the decision-making process. Decision Builder differs from other decision support technologies in that it streamlines the design process needed to build a DSS, through the application of visual thinking mechanisms. A DSS initiative under the Marble approach – and employing Marble tools – eliminates needless and costly steps in the design and implementation process, while maximizing accessibility, accuracy, and reliability (see Figure 3).

4.2 SELECTING AND DEPLOYING TOOLS

In the past, custom DSS developers built the user interface, all the algorithms, and the structured knowledge *from scratch* with only minimal assistance from existing off-the-shelf applications and software tools. This approach usually resulted in a "weak leg" in the DSS: either the user interface, or the algorithms, or the knowledge base fell short of user requirements.

FIGURE 3. Comparing methodologies: traditional DSS methodology (left) *versus* streamlined methodology using Decision Builder and the Marble approach (right)



Today, DSS designers can acquire a "bag of tricks" that includes everything from basic productivity software (such as spreadsheets) to specific niche tools (such as Decision Builder). These separate software sub-systems can be knit together via published interfaces, often referred to as Application Programming Interfaces (APIs). In this manner, DSS designers can leverage mature, well-tested algorithms (*e.g.*, those embedded in a spreadsheet or inference engine) without having to write, test, debug, and optimize them from scratch (see Figure 1).

This kind of tool integration reduces the amount of effort required to produce a custom DSS and allows more resources to be devoted to capturing the essence of competitive advantage: enterprise-wide *experience* and the *knowledge* gleaned from that experience.

4.2.1 Spotlight: the user interface to a custom DSS

The user interface to a decision support system is integral to its accessibility and effectiveness. In many DSS efforts of the past, the interface is foreign to the end user. It often consists of a Lisp-like shell that is usually powerful but difficult to use. If a DSS draws on familiar metaphors – such as dynamic bar graphs whose movements represent changing relationships among decision data, or gauges that represent mechanical gears, or knobs that represent real-time sensors – the end users will quickly be able to integrate the DSS into their existing environments.

Recently, user interface design tools such as NeXT Interface Builder™ have greatly simplified the task of constructing the appearance and behavior of custom applications. This new breed of tools provides "hooks" for integration with other tools, such as the Decision Builder tool kit. For example, the Decision Builder tool kit draws upon the rich user interface environment of Interface Builder through a *palette*. The palette (see below, the DB Palette) consists of pre-built widgets that obey the drag-and-drop, mouse-and-menu paradigm for DSS construction. In this manner, a DSS designer can knit together the user interface with the inference engine, primarily through point-and-click operations. The amount of programming declines, while the appearance and behavior of the interface dramatically improve.

4.3 PROTOTYPING

A DSS designer can deploy tools – libraries, objects, and other software sub-systems – to incorporate proprietary competitive knowledge into a DSS prototype. The prototype can be enhanced and expanded gradually and adapted (with minimal effort) to corporate internetworks. The DSS benefits through the prototype process by drawing on an ever-widening knowledge base, beginning with the local workgroup and expanding to the enterprise at large.

In addition, this approach relies on "living prototypes," as opposed to the throwaway prototypes common in many de-

velopment settings. A living prototype begins with a skeletal framework that accommodates new features, functions, and users in a controlled setting. The prototype evolves into a production-grade system – one that thrives in the internetwork environments of many companies – through a gradual, intensive process. In this approach, the investment in emerging DSS technology can be controlled, redirected, or suspended according to investment criteria and the demands of the business. The production system is the "finished prototype" and can be deployed immediately following the prototype process.

5.0 AN OVERVIEW OF THE DECISION BUILDER TOOL KIT

Marble's ***Decision Builder Tool Kit*** is a suite of software tools consisting of a graphical layout editor called ***Decision Builder***, a computational engine called ***Decision Maker***, an application programming interface for the engine called the ***Decision Builder API***, and assorted other development tools and example source code (see Figure 1). Decision Builder enables the creation of applications employing Bayesian Belief Network technology to calculate the probability of events. The Decision Builder layout editor allows knowledge bases for Bayesian reasoning systems to be constructed and updated in a fraction of the time required for text-based systems. Decision Maker contains two algorithms optimized for either precision or timely estimation of probabilities. The Decision Builder API is a public application programming interface that allows custom applications to tap the inference capabilities of Decision Maker.

5.1 BUILDING A DSS WITH DECISION BUILDER

A DSS developer develops and refines a knowledge base and prototypes the reasoning portion of a custom application from within the Decision Builder layout editor. When moving from prototype to production versions, the developer can add familiar graphic user interface metaphors (such as the gauges, dials and dynamic bar charts mentioned above) and links to other data and applications (*e.g.*, an enterprise SQL-based database that is accessed over the corporate internetwork; other custom applications; other conventional products such as spreadsheets). Many organizations are moving to distributed, cooperative processing environments – a migration addressed under the Marble Workgroup Computing Paradigm – with file/data/compute "servers" and desktop workstation "clients." The Decision Builder tool kit enables the kinds of solutions depicted in Figure 2, where the front-end to the DSS runs on some high-performance graphic workstation (such as a NeXT computer) and the computational engine, ***Decision Maker***, resides on the enterprise compute server (*e.g.*, a minicomputer, mainframe, or parallel supercomputer).

5.2 EXAMPLES OF TARGET APPLICATIONS

Decision Builder enables a new breed of computer-based decision aids since it recognizes the need for graphic interfaces, an open architecture, and networked operating (or production) environments. Using Decision Builder, developers might create command and control consoles for military purposes; medical decision aids; power management tools for utility companies; and executive support systems. These derivative systems can benefit from multimedia client workstations that present information using sound, color images, and even video.

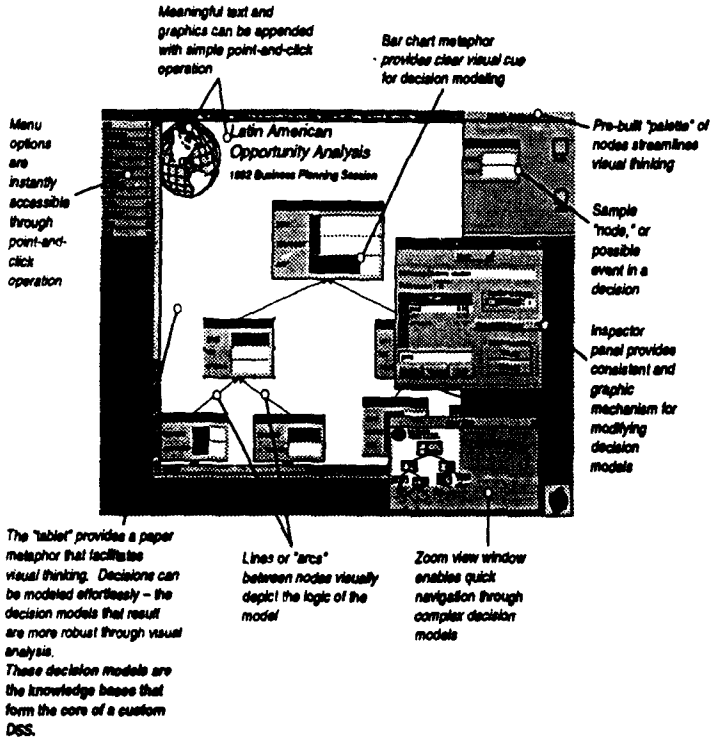
5.2.1 Spotlight: possible application to executive decision making

An executive user might draw on an Executive Support System – created with Decision Builder – for help in analyzing global expansion opportunities.

The Executive Support System in this instance might assist in selecting the appropriate avenue for global expansion, or for deciding when expansion into Latin American markets would be advantageous. Figure 4, below, depicts the knowledge base that would be created in a dynamic process, where the DSS developer and strategic planners from the Executive Committee model the decision process.

This visual model will serve as an industrial-strength *knowledge base* for the ESS.

FIGURE 4. Visual decision modeling and knowledge capture using the Decision Builder layout editor: examining global expansion opportunities



Other more direct and immediately useful visual metaphors, such as dynamic bar charts, or geographic maps displaying the flow of goods and services through Latin America, would be the interface mechanism for the executive user.

The executive user provides simple mouse-driven input. Transparent to the executive user, these mouse events trigger requests for the relevant data and decision support. Data flows from many points throughout the corporate internetwork and is consolidated and "crunched" by Decision Maker running on the corporate compute server (perhaps a mainframe or a parallel supercomputer; in many cases, another Unix-based workstation). The results are displayed on the executive's workstation using familiar visual metaphors (strategic use of color, sound, animation, *etc.*). The whole transaction would take seconds, depending on the complexity of the decision.

Most important, the executive receives decision making assistance from the ESS without having to worry about the intricacies of probabilistic expert systems, or the precise location of the data on the internetwork, or the form of the knowledge that went into creating the ESS with Decision Builder (*e.g.*, performance trend data for each geographic region; operating scenarios for international expansion opportunity). The executive might decide to charge the ESS with ongoing performance monitoring: it can then provide continual, real-time updates as new data becomes available, or as international currency rates fluctuate, or when new competitive threats or opportunities are identified.

Time sensitivity with accuracy – this is the pivotal advantage of any Executive Support System crafted with the Decision Builder tool kit and the Marble "tool integration" approach.

6.0 A TECHNICAL OVERVIEW OF THE DECISION BUILDER TOOL KIT

For experienced users of expert system technology, each module of the Decision Builder tool kit provides distinct features not available in other commercial offerings.

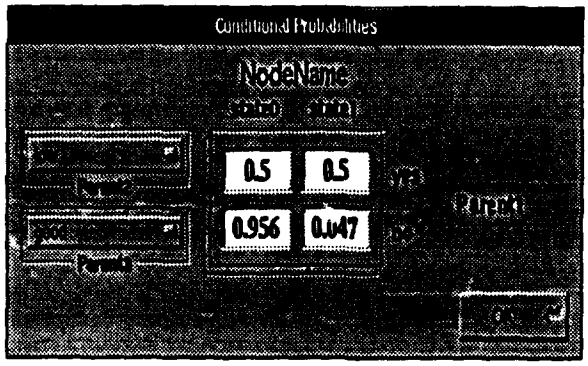
6.1 THE DECISION BUILDER LAYOUT EDITOR

The Decision Builder layout editor allows a user or knowledge engineer to create graphically a belief network and probability database by dragging and connecting nodes and probability distributions with a mouse (see Figure 4). In addition to its role as a belief network layout and interface design tool, Decision Builder allows the user to access the full power of Decision Maker, which contains the computational engine for belief networks. The Decision Builder layout editor can be used as a visual thinking tool and decision support system by itself, but the capability of Decision Maker also is available to custom applications and existing systems through the Decision Builder Application Programming Interface (see section 6.2, below).

The conditional probability distributions of nodes in a belief network and the connectivity of the nodes (events) in the network make up the knowledge base of a Bayesian inference system. Since people find it difficult to assign correct numerical probabilities for events that depend on several other events, Decision Builder provides several methods for modifying the probability distribution for a node. One method is appropriate when probabilities have been calculated from historical data (such as data from a

trend database) or when there are few events in the conditional distribution (see Figure 5).

FIGURE 5. Editing conditional probability distributions numerically using ***Decision Builder***

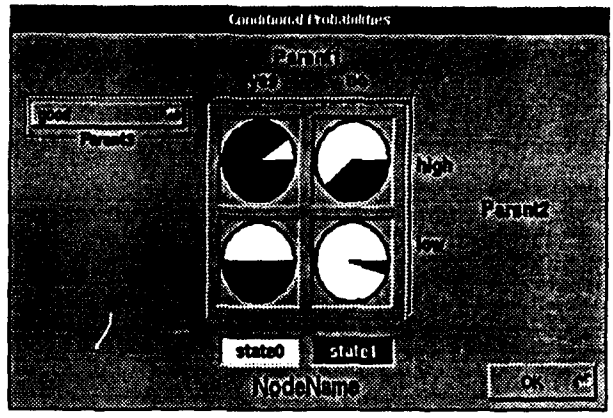


Two other ways to edit conditional probability distributions are graphic. These methods are more intuitive to most people and lead to more accurate distributions when the probabilities are being generated directly by experts. The pie wedges collapse one dimension of the probability space, so that more information about the relationships among events in the distribution can be presented in a single view (see Figure 6). A "stretchy histogram" view combines the best of pie wedge and textual mechanisms to clarify the decision process (events, connectivity and conditional probabilities) even further.

A fourth method for manipulating conditional probability distributions is provided by the Decision Builder Improv Integration Kit[2], which contains an Improv module allowing probability distributions to be manipulated as multidimensional spreadsheets. This method can allow an entire distribution for a node with up to twelve conditional events to be displayed in a single view.

Finally, an user may create her own probability manipulation paradigm by making calls to the appropriate Decision Builder API library functions.

FIGURE 6. Editing conditional probability distributions graphically using ***Decision Builder***



6.2 THE DECISION BUILDER API

All functions of Decision Maker are available to application developers through the Decision Builder API. Once a knowledge base has been created with the Decision Builder layout editor, the custom DSS can make full use of the knowledge base over a (TCP/IP-based) network via the Decision Builder API. A complete application for an end user can often consist of simply the belief network created by the layout editor, a license for Decision Maker, and the custom software product that triggers the API to make inferences with the knowledge contained in the belief network.

The Decision Builder API makes Decision Builder an open tool kit that any other computer application can use. For instance, Lotus Improv provides an API that can be used in the "tool integration" approach, as suggested above (see Figure 1). In this manner, an Improv spreadsheet interfaces easily with Decision Builder through their respective APIs.

Entire decision support systems employing Improv financial models, Decision Builder belief networks, and links to SQL databases can be created with the Decision Builder API. Such systems can become "intelligent spreadsheets," providing advice and up-to-the-moment financial (or other analytical) information.

2. Contact Marble directly for information about expected ship dates for the Improv Integration Kit.

6.3 DECISION MAKER

Decision Maker is based on the principles that algorithms for decision support should (a) be appropriate to task demands and (b) draw upon provably correct mathematical techniques (such as Bayes's theorem) rather than obscure heuristics. To realize these principles Marble chose probabilistic reasoning in Bayesian belief networks as the basis for all the inference algorithms in the system. Marble then developed two complementary algorithms optimized for a different range of *size-and-speed* versus *accuracy* constraints. Decision Maker can operate with either of the two algorithms, both of which draw upon the *same* knowledge base, and each of which posits strengths for particular classes of problems.

6.3.1 The Lauritzen-Spiegelhalter algorithm: a precision approach

Marble's implementation of the Lauritzen-Spiegelhalter algorithm, also known as probability propagation in trees of cliques, is deterministic. It produces exact answers in little time for belief networks of moderate size, but like all deterministic algorithms for belief network calculation, it experiences exponential time complexity. This algorithm gives little information until computation terminates, making it inappropriate for large networks in time-critical settings.

6.3.2 The BN-RAS algorithm: an estimation approach

The second algorithm supported by Decision Maker is an enhanced version of the BN-RAS estimation algorithm for belief networks. BN-RAS is a randomized approximation scheme employing Markov Sampling to produce approximate answers for large networks in a small fraction of the time required by the Lauritzen-Spiegelhalter algorithm. It complements the Lauritzen-Spiegelhalter algorithm since it is targeted at complex decisions (large multiply-connected Bayesian Belief networks) in time-critical settings. The BN-RAS algorithm produces a first estimate *immediately* and continues to give feedback as the accuracy of estimates approaches certainty over time.

7.0 CONCLUSION

As many business pundits have observed, information technology has been at once a blessing and a curse: a blessing because it has placed so much useful information at the user's fingertips and a curse because it has placed so much useful information at the user's fingertips.

Sorting through these masses of information and eliminating information overload is a heady undertaking. However, an organization's competitive advantage depends on its ability to tap relevant information and shape it into knowledge that influences mission critical decisions.

Two issues emerge as decision makers across industries turn to custom software systems to support mission critical decisions: *accuracy* and *speed*. They cannot wait forever to enjoy the fruits of their investments, nor can they suffer from inaccuracy when they deploy custom decision support systems. The Decision Builder tool kit, when coupled with the Marble's Workgroup Computing paradigm and "tool integration" approach, enables rapid design, prototype and delivery of decision aids that meet new challenges.

8.0 ABOUT MARBLE...

Marble Associates, Inc., is a leading technology strategy and consulting organization specializing in business re-engineering using the ***Marble Workgroup Computing Paradigm***. The Workgroup Computing Paradigm positions Unix workstations along with departmental or divisional servers, mainframes, network topologies and protocols, *etc.*, in a distributed, enterprise-wide model of technology systems and data flows. We have significant expertise in integrating open systems technology into heterogeneous environments, including traditional mainframe-class environments.

Marble Software Products, a division of Marble Associates, Inc., is dedicated to building high quality, niche software products that subscribe to the Marble Workgroup Computing Paradigm. Marble Software Products is staffed by expert software engineers who work closely with the consulting staff at Marble Associates to identify niche product opportunities that come up repeatedly in Marble's consulting engagements. Marble Software Products currently supports the following products for NeXT and other Unix computers:

- ***Marble Teleconnect;***
- ***Designer Labels;***
- ***Mark-Sense Template Builder*** (for MS-DOS).

Please write or call for more information and other Marble White Papers.

Ray Bloom, Vice-President
Marble Software Products,
a division of Marble Associates, Inc.
1641 North First Street
San Jose, California 95112
phone: (408) 436–7299
fax: (408) 436–7147
e-mail: ray@marble.com

Rick Chavez, Vice-President
Marble Associates, Inc.
38 Edge Hill Road
Waltham, Massachusetts 02154
phone: (617) 891-5555
fax: (617) 891-5226
e-mail: rick@marble.com